

CLAIMS

What is claimed is:

1 1. A method comprising:
2 performing identification and decomposition of fundamental units performed by
3 underlying hardware to process a packet;
4 forming software objects by abstracting the fundamental units to packet processing
5 into the software objects, such that the software objects encapsulate and represent
6 functionality of the fundamental units; and
7 creating an object-oriented programming model using the software objects, such
8 that the software object modules enable programming of underlying hardware to process
9 packets as programmed using the object-oriented programming model.

1 2. The method of claim 1, further comprising:
2 selecting one or more software objects from the object-oriented programming
3 model;
4 programming the one or more selected software objects models to perform a desired
5 packet processing functionality;
6 connecting the one or more programmed software objects to form a directed graph
7 of packet flow to complete definition of the desired packet processing functionality; and
8 directing underlying hardware loaded with the directed graph of programmed
9 software objects to process packets in accordance with the desired packet processing
10 functionality.

1 3. The method of claim 2, wherein the connecting the one or more
2 programmed software objects further comprises:
3 selecting a first stage object as an input port of the directed graph to direct a physical
4 interface and packet framing;
5 selecting a second stage object coupled to the first stage object to direct filtering and
6 matching algorithms on packets;

7 selecting a third stage object coupled to the second stage object to direct routing of
8 packets to one or more third stage object outputs;
9 selecting a fourth stage object coupled to an output from the one or more third stage
10 object outputs to direct gathering of statistical information regarding packets and packet
11 flows;
12 selecting an fifth stage object coupled to an output from the one or more third object
13 outputs to direct packet modification;
14 selecting a sixth stage object coupled to an output from the one or more third object
15 outputs to direct packet routing to a sixth stage object output; and
16 selecting a final stage object as an output port of the directed graph, the final stage
17 object coupled to the sixth object output.

1 4. The method of claim 2, wherein the directing underlying hardware further
2 comprises:

3 making requests from the software objects contained in the direct graph to
4 underlying packet forwarding hardware in accordance with the desired packet processing
5 functionality; and

6 performing packet processing by the packet forwarding hardware in response to the
7 software object requests, such that the directed graph of software objects control packet
8 data flow through the packet forwarding hardware.

1 5. The method of claim 1, wherein the forming the software objects further
2 comprises:
3 using as the one or more software objects
4 a first stage object to define a physical interface and packet framing,
5 a second stage object to direct filtering and matching algorithms on packets,
6 a third stage object to direct packet flow policy,
7 a fourth stage object to direct packet routing to scatterer object outputs,
8 a fifth stage object to direct packet collecting and routing scattered packets;
9 an sixth stage object to direct packet modification,
10 a seventh stage object to direct packet flow policy, and
11 an eighth stage object to direct gathering of statistical information regarding
12 packets and packet flows.

1 6. The method of claim 5, wherein the one or more software objects each
2 include one or more inputs and one or more outputs enabling formation of compositions of
3 objects sharing a common interface to direct packets processing as a group of software
4 objects and forming directed graphs of software objects to direct packet data flow through
5 the packet forwarding hardware.

1 7. A machine-readable medium having stored thereon data representing
2 sequences of instructions, the sequences of instructions which, when executed by a
3 processor, cause the processor to performing a method comprising:
4 performing identification and decomposition of fundamental units performed by
5 underlying hardware to process a packet;
6 forming software objects by abstracting the fundamental units to packet processing
7 into the software objects, such that the software objects encapsulate and represent
8 functionality of the fundamental units; and

creating an object-oriented programming model using the software objects, such that the software object modules enable programming of underlying hardware to process packets as programmed using the object-oriented programming model.

8. The machine-readable medium of claim 7, further comprising:
selecting one or more software objects from the object-oriented programming model;
programming the one or more selected software objects models to perform a desired packet processing functionality;
connecting the one or more programmed software objects to form a directed graph of packet flow and complete definition of the desired packet processing functionality; and
directing underlying hardware loaded with the directed graph of programmed software objects to process packets in accordance with the desired packet processing functionality.

9. The machine-readable medium of claim 8, wherein the connecting the one or more programmed software objects further comprises:
selecting a first stage object as an input port of the directed graph to direct a physical interface and packet framing;
selecting a second stage object coupled to the first stage object to direct filtering and matching algorithms on packets;
selecting a third stage object coupled to the second stage object to direct routing of packets to one or more third stage object outputs;
selecting a fourth stage object coupled to an output from the one or more third stage object outputs to direct gathering of statistical information regarding packets and packet flows;
selecting an fifth stage object coupled to an output from the one or more third object outputs to direct packet modification;

14 selecting a sixth stage object coupled to an output from the one or more third object
15 outputs to direct packet routing to a sixth stage object output; and
16 selecting a final stage object as an output port of the directed graph, the final stage
17 object coupled to the sixth object output.

1 10. The machine-readable medium of claim 8, wherein the directing underlying
2 hardware further comprises:

3 making requests from the software objects contained in the direct graph to
4 underlying packet forwarding hardware in accordance with the desired packet processing
5 functionality; and

6 performing packet processing by the packet forwarding hardware in response to the
7 software object requests, such that the directed graph of software objects control packet
8 data flow through the packet forwarding hardware.

9 11. The machine-readable medium of claim 7, wherein the forming the software
10 objects further comprises:

11 using as the one or more software objects

12 a first stage object to define a physical interface and packet framing,

13 a second stage object to direct filtering and matching algorithms on packets,

14 a third stage object to direct packet flow policy,

15 a fourth stage object to direct packet routing to scatterer object outputs,

16 a fifth stage object to direct packet collecting and routing scattered packets;

17 a sixth stage object to direct packet modification,

18 a seventh stage object to direct packet flow policy, and

19 an eighth stage object to direct gathering of statistical information regarding
20 packets and packet flows.

1 12. The machine-readable medium of claim 11, wherein the one or more
2 software objects each include one or more inputs and one or more outputs enabling
3 formation of compositions of objects sharing a common interface to direct packets
4 processing as a group of software objects and forming directed graphs of software objects
5 to direct packet data flow through the packet forwarding hardware.

1 13. An apparatus, comprising:
2 a processor; and
3 a memory coupled to the processor and including software object linked together to
4 from a directed graph of packet flow to direct the processor to perform packet processing in
5 a manner specified using an object-oriented programming model and compiled to produced
6 the directed graph of software objects.

1 14. The apparatus of claim 13, wherein the memory is further configured to
2 relay requests from the software objects contained in the direct graph to the processor, such
3 that processor performs packet processing in response to the software object requests for
4 control packet data flow through the processor.

1 15. The apparatus of claim 13, wherein the processor is configured as a network
2 processor.

1 16. The apparatus of claim 13, wherein the processor is configured as a an
2 application specific integrated circuit.

1 17. A system comprising:
2 a wide area network;
3 a local area network; and
4 a processor coupled between the wide area network and the local area network on to
5 form a network, the processor having
6 a memory coupled to the processor and including software objects linked
7 together to form a directed graph of packet flow to direct the processor to perform
8 packet processing in a manner specified using an object-oriented programming
9 model and compiled to produce the directed graph of software objects.

1 18. The system of claim 17, wherein the memory is further configured to relay
2 requests from the software objects contained in the directed graph to the processor, such
3 that processor performs packet processing in response to the software object requests for
4 controlling packet data flow through the processor.

1 19. The system of claim 17, wherein the processor is configured as a network
2 processor.

1 20. The system of claim 17, wherein the processor is configured as an
2 application specific integrated circuit.

1 21. The system of claim 17, wherein each software object within the directed
2 graph performs a data-path packet processing task functionality, such that the directed
3 graph of software objects performs a plurality of data-path packet procession tasks with a
4 single device.